W1258

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-353284
(43)Date of publication of application : 24.12.1999

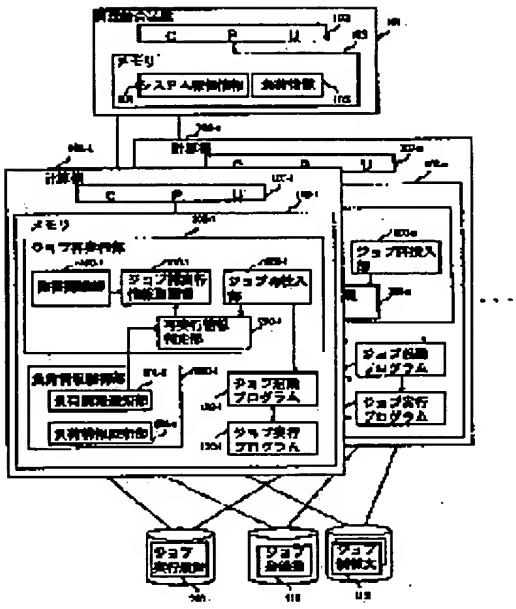
(51)Int.Cl. G06F 15/16
G06F 11/14
G06F 11/16

(21)Application number : 10-161808
(22)Date of filing : 10.06.1998

(71)Applicant : HITACHI LTD
(72)Inventor : HONDO YURI
NAGASUGA HIROFUMI
AKIBA CHIEKO
IWAKURA YOSHIYUKI

(54) JOB RE-EXECUTING METHOD

(57)Abstract:
PROBLEM TO BE SOLVED: To attain the efficient operation of a cluster type computer system wherein the influence of computer faults is suppressed by finding a job to be re-executed at a local computer node among jobs which were executed on a 1st computer and re-executing the found job on the local computer node.
SOLUTION: Computers 106 connected by a high-speed coupling device 101 consisting of a CPU 102 and a memory 103 are each composed of a CPU 107 and a memory 108. In a memory 108 of each computer 106, a jog start program 109 which starts a job in response to a report from a jog reloading part 600 and a job execution program 110 which executes the job are set. A computer which re-executes the jog is determined according to information of the job and load information from a node. When it is decided that the determined local computer suitably re-executes the job, the job is loaded onto the local computer and re-executed.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

W1258

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（A）　(11)特許出願公開番号

特開平11－353284

(43)公開日　平成11年(1999)12月24日

(51)Int.Cl.[6]		識別記号	FI		
G06F	15/16		G06F	15/16	420J
	11/14	310		11/14	310Z
	11/16	310		11/16	310Z

審査請求　未請求　請求項の数4　OL　（全 9 頁）

(21)出願番号　特願平10－161808

(22)出願日　平成10年(1998)6月10日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地

(72)発明者　本堂　友理
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72)発明者　長須賀　弘文
神奈川県川崎市麻生区王禅寺1099番地　株式会社日立製作所システム開発研究所内

(72)発明者　秋葉　千江子
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内

(74)代理人　弁理士　小川　勝男

最終頁に続く

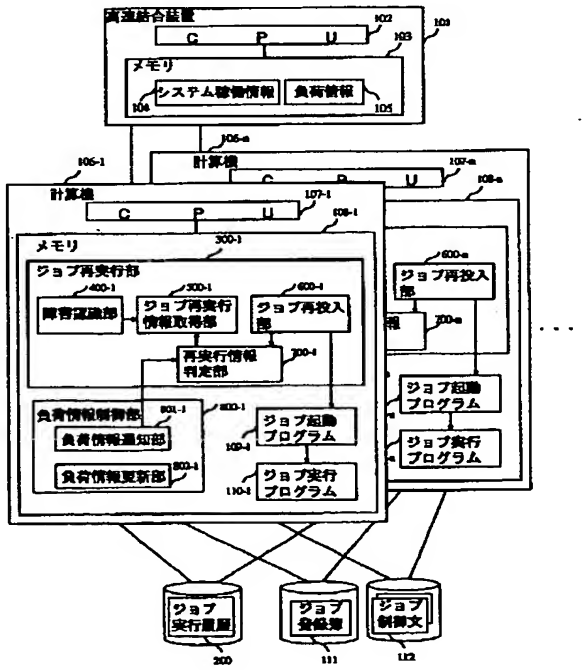
(54)【発明の名称】　ジョブ再実行方法

(57)【要約】

【課題】複数の計算機から構成される計算機システムにおける計算機障害時にジョブを再実行させる場合の、負荷分散機能を用いた効率的なジョブ再実行技術を提供する。

【解決手段】障害の発生した計算機で実行中であったジョブを再実行させる際に、再実行を行う候補となる計算機の負荷情報と、被再実行ジョブの中断されるまでの実行状況の情報から、再実行させる計算機を判定することで効率的な実行と計算機障害の影響の早期解決を実現する。

図1



【特許請求の範囲】
【請求項１】複数の計算機から構成される計算機システムに、上記計算機システム上で動作している全てのジョブが使用する計算機資源と稼働状況に関する情報を収集し、上記システム内の全ての計算機ノードが上記収集した情報を参照することを可能とするステップを有する計算機システムのジョブの再実行方法において、
第一の上記計算機ノードに障害が発生した場合に、上記計算機システム内の当該第一の計算機を除いた計算機から成る第二の計算機群の各々は、上記収集した情報から上記第一の計算機上で実行していたジョブの特性を解析し、判断基準に基づいて自計算機ノードで再実行させるジョブを上記第一の計算機上で実行していたジョブの中から求め、求めたジョブを自計算機ノード上で再実行させるステップを有することを特徴とするジョブ再実行方法。
【請求項２】請求項１記載のジョブ再実行方法において、上記判定基準としてＣＰＵ使用率を用い、自計算機のＣＰＵ負荷が他計算機に対して低く、自計算機のＣＰＵ負荷から得るＣＰＵ使用率の閾値を越えないＣＰＵ使用率であることが上記収集した情報から得られるジョブを求めて再実行を行うことを特徴とするジョブ再実行方法。
【請求項３】請求項１記載のジョブ再実行方法において、上記判定基準として上記中断したジョブがＣＰＵを多く使用するジョブであるか又は入出力処理を多く行うジョブであるかを用い、自計算機のＣＰＵ負荷が低い場合はＣＰＵを多く使用するジョブを求め、自計算機の入出力装置の使用率が低い場合は入出力処理を多く行うジョブを求めて再実行を行うことを特徴とするジョブ再実行方法。
【請求項４】請求項１記載のジョブ再実行方法において、上記判定基準として上記中断したジョブの中断するまでの実行時間を用い、自計算機の予め指定された閾値を越えない実行時間を持つジョブを求めて再実行を行うことを特徴とするジョブ再実行方法。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】本発明は、複数計算機からなり、各々の計算機の負荷情報が相互に取得可能な手段を有する計算機システムでのジョブ再実行方法に関し、特にジョブ実行中の計算機の障害発生時におけるジョブ再実行方法に関するものである。
【０００２】
【従来の技術】大規模システムでは、その社会的用途から高性能及び高信頼性が求められており、その要求から生まれた機能の一つに、システム障害時におけるジョブの自動再実行機能がある。複数計算機から成る計算機システムにおいて、ある計算機に障害が発生し実行継続が不可能となったジョブの再実行方式の一例として特開平７－１７５７６６号公報の「疎結合多重システムのジョブ再実行制御方式」がある。
【０００３】これは、ジョブを投入する場合は予め再実行させる計算機を定義しておき、ある計算機に障害が発生した場合は、その計算機の回復を待たずに、上記の予め定義しておいた別計算機にスケジューリングすることで、速やかに再実行が行えるような手段を提供するものである。
【０００４】
【発明が解決しようとする課題】大規模システムでは、複数の計算機をノードとして接続することで構成されるクラスタ型計算機システム上で業務実施するのが主流になりつつある。このようなクラスタ型計算機システム上で効率よく業務処理を遂行するためには、各ノードの計算機資源を有効に活用する必要がある。そのためには、特定の計算機ノードに業務処理が集中することがないように、各計算機ノードに処理を分散させている。
【０００５】上記従来技術は、障害により再実行する必要の生じたジョブをスケジュールする計算機は、予めジョブ制御言語や特定の情報格納領域に記述されていることが前提となっている。従って、上記のようなクラスタ型計算機システムで上記従来技術を実現した場合は、障害が発生した場合は特定の計算機に処理が集中し、効率よい業務処理遂行が妨げられる可能性がある。
【０００６】また、上記従来技術では、障害が発生した場合には、特定の計算機がマスタ計算機としてジョブの再スケジューリングを行うことになっている。従って、そのマスタ計算機自体に障害が発生した場合は回復作業に支障をきたす可能性がある。上記クラスタ型計算機システムは、従来のマスタスレーブ構成をとる疎結合マルチプロセサシステムとは異なり、特定の計算機をマスタとして固定することなく自律的に処理を行い耐障害性を高めることを特徴とするシステムである。従って固定的な特定のマスタ計算機の存在を前提とする上記従来技術ではこのクラスタ型計算機システムの利点を損なう恐れがあった。
【０００７】本発明は複数計算機から成りシステムを構成する各計算機の負荷情報を取得する手段を有する計算機システムにおけるジョブの再実行方法に関するものであり、以下の二つの目的を持つ。
【０００８】本発明の第一の目的は、この負荷情報取得手段を利用し、再実行するべきジョブの特性をふまえた再スケジューリングすることで、計算機障害の影響を抑えた効率的なクラスタ型計算機システムの運用を実現することにある。
【０００９】本発明の第二の目的は、複数の計算機ノードがジョブの回復手段を有することで、特定の計算機ノードの障害によるジョブ再実行不能状態を回避し、システム全体の耐障害性を向上させることにある。
【００１０】

【課題を解決するための手段】効率よく業務処理を遂行するためには、特定の計算機ノードに業務処理が集中することがないように、各計算機ノードに処理を分散させなければならない。そのために、クラスタ型計算機システムでは各々の計算機の負荷を収集し、その情報をシステム全体の処理状況として各計算機ノードで実行される処理が取得できる負荷情報取得手段が具備されているものが多い。本発明ではこの負荷情報取得手段を利用し、再実行するべきジョブをその特性に見合った計算機に再スケジューリングすることで、計算機障害の影響を抑えたクラスタ型計算機システムの運用を実現する。

【００１１】ジョブを実行する複数の計算機とそれらの負荷情報等を格納するために上記複数の計算機各々から書き込み・参照可能な領域を保持する１つの高速結合装置、または共有拡張記憶装置、または特定計算機の記憶装置、またはその他の記憶装置から成り、上記複数の計算機が自計算機の負荷情報を収集しシステム全体の負荷情報を格納する領域に対する更新を行う負荷情報更新処理機能と、上記格納された負荷情報を要求に応じて通知する負荷情報通知処理機能からなる負荷情報制御手段を有し、ジョブの実行内容を定義したジョブ制御文を格納した記憶装置と、ジョブの実行順序を登録するジョブ登録簿を格納した記憶装置と、各計算機で実行しているジョブの実行履歴を格納した記憶装置がシステム内の計算機各々から共通して参照及び更新可能である計算機システムで、上記第一の目的を達成するために、上記各計算機からアクセス可能な領域におかれた計算機システムの稼働情報からシステム内の計算機に発生した障害を検知するステップと、上記検知ステップにおいて障害が発生した場合は、発生した計算機とそこで実行されていたジョブの情報を取得し、そのジョブを再実行する準備をするステップと、上記準備をするステップの通知により、障害の発生した計算機での実行履歴から再実行するジョブの情報を取得するステップと、ジョブを再実行する際のシステムの各計算機ノードの負荷情報を取得するステップと、上記ジョブの情報と計算機ノードの負荷情報から、上記ジョブを再実行する計算機を決定するステップと、上記決定により自計算機が上記ジョブを再実行するに適した計算機であると判定された場合は、そのジョブを自計算機に再投入し実行を行うステップを設ける。ここで、上記障害の発生した計算機での実行履歴から得るジョブの情報は、ジョブの中断までの実行時間、実行時間に占めるＣＰＵ使用時間、前記ＣＰＵ使用時間と前記実行時間の比率、使用した入出力装置台数等が挙げられる。

【００１２】上記第二の目的を達成するために、上記計算機システム内の各々の計算機が、上記第一の目的を達成するための上記複数のステップを具備し、ある計算機に障害が発生した場合は、他の計算機上の上記複数のステップが実行されるものとする。

【００１３】上記各計算機からアクセス可能な領域におかれた計算機システムの稼働情報からシステム内の計算機に発生した障害を検知するステップと、上記検知ステップにおいて障害が発生した場合は、発生した計算機とそこで実行されていたジョブの情報を取得し、そのジョブを再実行する準備をするステップにより、ジョブの再実行を行う計算機が自律的に中断したジョブの再投入を行うことができる。

【００１４】上記準備をするステップの通知により、障害の発生した計算機での実行履歴から再実行するジョブの情報を取得するステップと、ジョブを再実行する際のシステムの各計算機ノードの負荷情報を取得するステップにより、自計算機で再実行するためのジョブの情報と、計算機システム内での自計算機の負荷情報を把握するための情報を取得することができる。

【００１５】上記ジョブの情報とノードの負荷情報から、上記ジョブを再実行する計算機を決定するステップと、上記決定により自計算機が上記ジョブを再実行するに適した計算機であると判定された場合は、そのジョブを自計算機に再投入し実行を行うステップにより、再実行すべきジョブの特性に合わせた計算機での再実行が可能となる。

【００１６】

【発明の実施の形態】本発明の実施形態を図を用いて詳細に説明する。

【００１７】まず第一の実施形態について図１～図４を用いて説明する。

【００１８】本実施形態では、一つ以上の複数の計算機により構成されるシステムを対象とする。複数の計算機は相互に通信を行う手段として、各計算機から更新・参照可能である領域を保持する高速結合装置により各計算機が接続されているものとする。また、ジョブの実行内容を定義したジョブ制御文を格納した記憶装置と、ジョブの実行順序を登録するジョブ登録簿を格納した記憶装置と、各計算機で実行しているジョブの実行履歴を格納した記憶装置が各計算機から共通して更新・参照可能であるものとする。

【００１９】図１は、本実施形態の基本的な構成を表している。ＣＰＵ１０２およびメモリ１０３から構成される高速結合装置１０１により接続された複数の計算機１０６は各々がＣＰＵ１０７およびメモリ１０８から構成される。

【００２０】システム内の各計算機の負荷情報を管理する負荷情報制御部８００は計算機１０６のメモリ１０８上に置かれ、負荷情報更新部８０２と負荷情報通知部８０１から構成される。負荷情報更新部８０２は一定時間毎に自計算機の負荷情報を収集し、その内容を高速結合装置１０１のメモリ１０３上の負荷情報１０５の領域に書き込む処理を行う。負荷情報通知部８０１は、本発明のジョブ再実行部３００やその他のプログラムの要求に

より、負荷情報１０５の情報を通知したり、負荷情報１０５の内容から最も負荷の低い計算機を判定し結果を通知したりする処理を行う。

【００２１】各計算機１０６は自計算機が稼働しているという情報をシステム稼働情報１０４に置き、他計算機に障害が発生したか否かという情報を取得できるようになっている。

【００２２】各々の計算機１０６のメモリ１０８上には障害により中断されたジョブを再実行するジョブ再実行部３００が設置されている。ジョブ再実行部３００は、システム稼働情報１０４から他計算機の障害を認識する障害認識部４００と、再実行すべきジョブの情報をジョブ実行履歴２００とジョブ制御文１１２から取得して再実行の準備を行うジョブジョブ再実行情報取得部５００と、取得したジョブの情報と負荷情報通知部８０１から取得した負荷情報１０５からジョブの特性を判定し、再実行するのに適した計算機を選択する再実行情報判定部７００と、再実行するのに適した計算機が自計算機であった場合は自計算機に対してジョブの再投入を行うジョブ再投入部６００から構成される。ここでは、中断したジョブの情報として、そのジョブの中断するまでの平均ＣＰＵ使用率を用いて説明を行う。

【００２３】各々の計算機１０６のメモリ１０８上には、ジョブ再投入部６００からの通知でジョブの起動を行うジョブ起動プログラム１０９と、ジョブの実行を行うジョブ実行プログラム１１０が設置されている。

【００２４】ジョブ実行履歴２００の構造を図２に示す。ジョブ実行履歴２００は計算機１０６のジョブ実行単位である空間に対応した複数のレコードから構成される。その空間で実行されていたジョブ名、実行を開始した日付、開始した時刻、そのジョブの障害が発生した時刻までのＣＰＵ使用時間等が含まれている。

【００２５】障害により実行が中断されたジョブの再実行を行うジョブ再実行部３００の処理の流れを図３により説明する。

【００２６】他計算機に障害が発生したか否かの情報を取得し、障害が発生した場合は他計算機で実行が中断されたジョブの回復処理を開始する（ステップ４００）。

【００２７】障害の発生した計算機の障害が発生した時刻等の情報を含むシステム稼働情報１０４と、障害の発生した計算機で実行されていたジョブの情報を含むジョブ実行履歴２００と、ジョブの処理内容を定義したジョブ制御文１１２からジョブの情報を取得する（ステップ５００）。

【００２８】再実行すべきジョブが存在するかを判定する（ステップ３０１）。上記判定が偽であった場合、処理を終了する。上記判定が真であった場合、取得した情報から再実行情報判定処理７００により、再実行するに適した計算機を求める（ステップ３０２）。

【００２９】上記判定から得られた再実行に適した計算機が自計算機であるか判定を行う（ステップ３０３）。上記判定が偽であった場合、ステップ３０４から処理を行う。上記判定が真であった場合、自計算機のジョブ起動プログラム１０９に対し、ジョブの起動要求を行う（ステップ６００）。

【００３０】まだ再実行すべきジョブがあるかを判定する（ステップ３０４）。上記判定が真であった場合、ステップ３０２から次の処理を行う。上記判定が偽であった場合、処理を終了する。

【００３１】再実行情報判定部の処理の流れを図４により説明する。

【００３２】ジョブ再実行プログラムから再実行すべきジョブの情報を取得する。システム稼働情報１０４に含まれる障害が発生した時刻の情報と、ジョブ実行履歴２００に含まれるジョブが実行開始した時刻の情報から、中断するまでの経過時間が取得できる。また、ジョブ実行履歴２００からそのジョブの中断するまでのＣＰＵ使用時間が取得できる。ここから対象となるジョブの平均したＣＰＵ使用率が求められる（ステップ７０１）。

【００３３】次に負荷情報通知部８０１から、再実行するジョブが実行可能な計算機の現時点での負荷情報を取得する（ステップ７０１）。

【００３４】ステップ７０１で求めたジョブが消費するＣＰＵ使用率を割り当て可能な計算機を判定する（ステップ７０３）。

【００３５】ステップ７０３により求めたジョブを再実行するのに適した計算機の判定結果を戻す（ステップ７０４）。

【００３６】以上、第一の実施形態を具体的に説明したが、前記実施形態において再実行するジョブから得られる情報をジョブの中断するまでの平均ＣＰＵ使用率としたが、これを中断するまでのジョブの実行時間としてもよい。その場合は、長大な実行時間がかかるジョブを特定計算機に分担させるといったシステムの形態が実現できる。

【００３７】また、再実行するジョブから得られる情報を、ジョブ制御文１１２から得られる入出力装置の台数としてもよい。その場合は、入出力装置の台数に比較的余裕のある計算機に振り分けることが可能となる。

【００３８】以上、本発明の実施形態において各計算機が接続される装置を高速結合装置として説明したが、その装置を共有拡張記憶装置としてもよい。その場合の基本的な構成を図５に示す。システムを構成する全ての計算機１０６－ｎと接続されている共有拡張記憶装置１１３上にシステム稼働情報１０４と負荷情報１０５が設置されるものとする。

【００３９】また、上記の領域を保持する装置をディスク装置としてもよい。その場合の基本的な構成を図６に示す。システムを構成する全ての計算機１０６－ｎと接続されているディスク装置１１４上にシステム稼働情報

１０４と負荷情報１０５が設置されるものとする。

【００４０】

【発明の効果】本発明によれば、システム内の計算機の障害によって再実行しなければならないジョブが短時間に大量に発生しても、効率的に処理できるようにジョブの特性にあわせてシステム内に分散させ再実行することができる。

【００４１】さらに、障害のため実行が中断したジョブを再実行するため分散させる機能を特定計算機に偏らせることなく、その時の計算機システムの運用状況に合わせて分散することが可能となり、高性能化、高信頼化できる。

【図面の簡単な説明】

【図１】本発明のジョブ再実行方法の第一の実施例を示した計算機システムの構成図である。

【図２】本発明の入力情報となるジョブ実行履歴の説明図である。

【図３】ジョブ再実行部の処理を説明したフローチャートである。

【図４】再実行情報判定部の処理を説明したフローチャートである。

【図５】本発明のジョブ再実行方法を実現した計算機システムの別の構成図である。

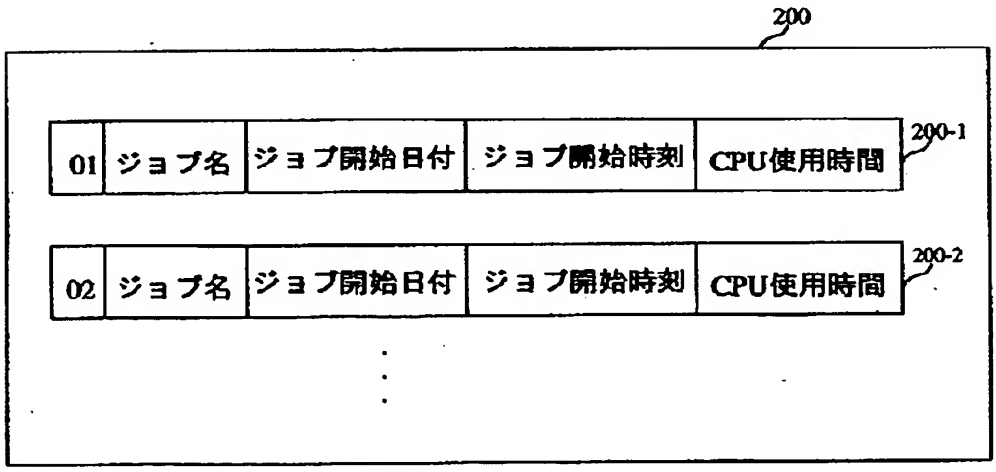
【図６】本発明のジョブ再実行方法を実現した計算機システムの別の構成図である。

【符号の説明】

１０１…高速結合装置、　１０２…高速結合装置のＣＰＵ、１０３…高速結合装置のメモリ、１０４…システム稼働情報管理テーブル、１０５…負荷情報、１０６－ｎ…計算機、１０７－ｎ…計算機のＣＰＵ、１０８－ｎ…計算機のメモリ、３００－ｎ…再実行部、４００－ｎ…障害認識部、５００－ｎ…ジョブ再実行情報取得部、６００－ｎ…ジョブ再投入部、７００－ｎ…再実行情報判定部、８００－ｎ…負荷情報制御部、１０９－ｎ…ジョブ起動プログラム、１１０－ｎ…ジョブ実行プログラム、２００－ｎ…ジョブ実行履歴、１１１－ｎ…ジョブ登録簿、１１２－ｎ…ジョブ制御文。

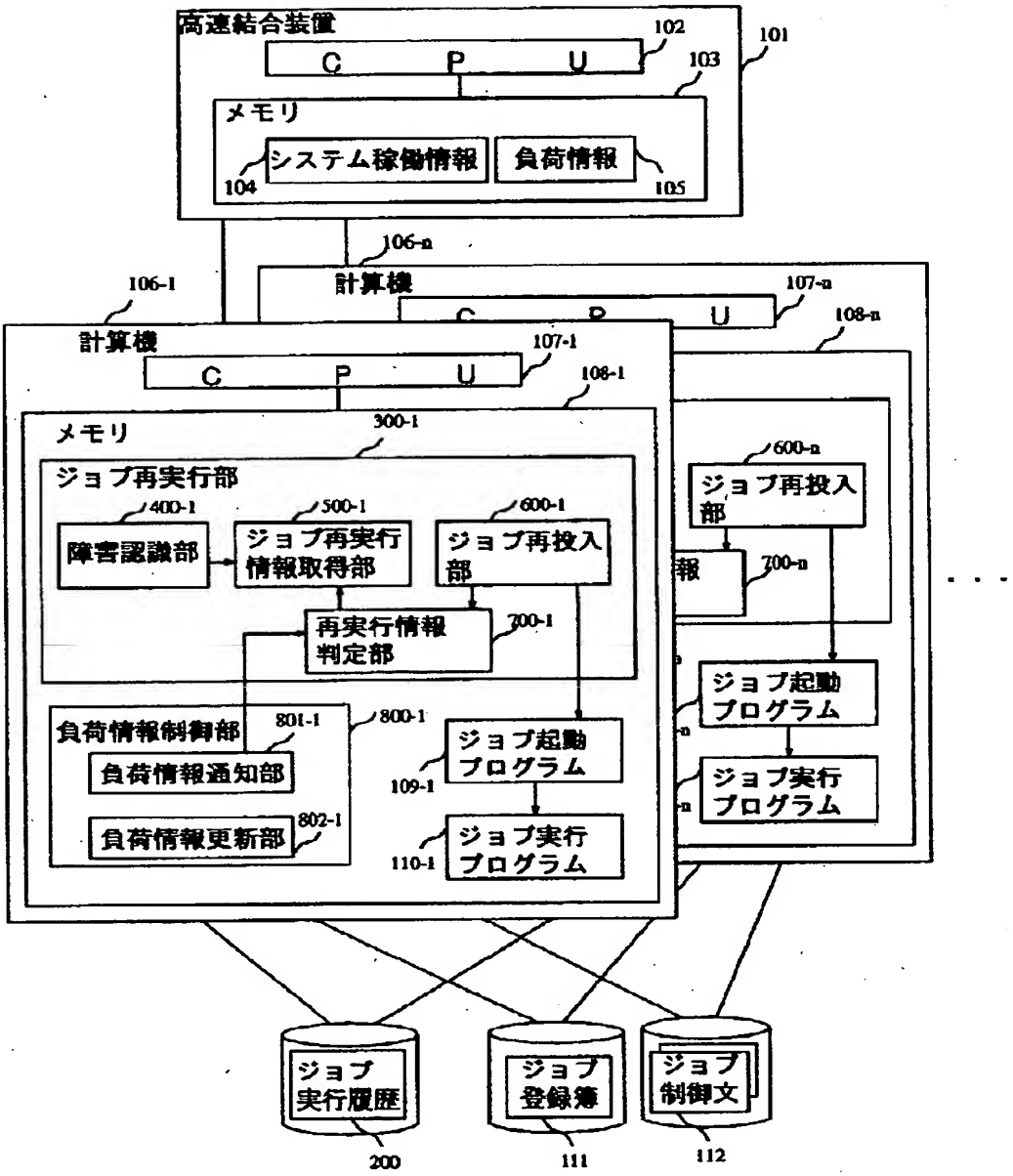
【図２】

図２



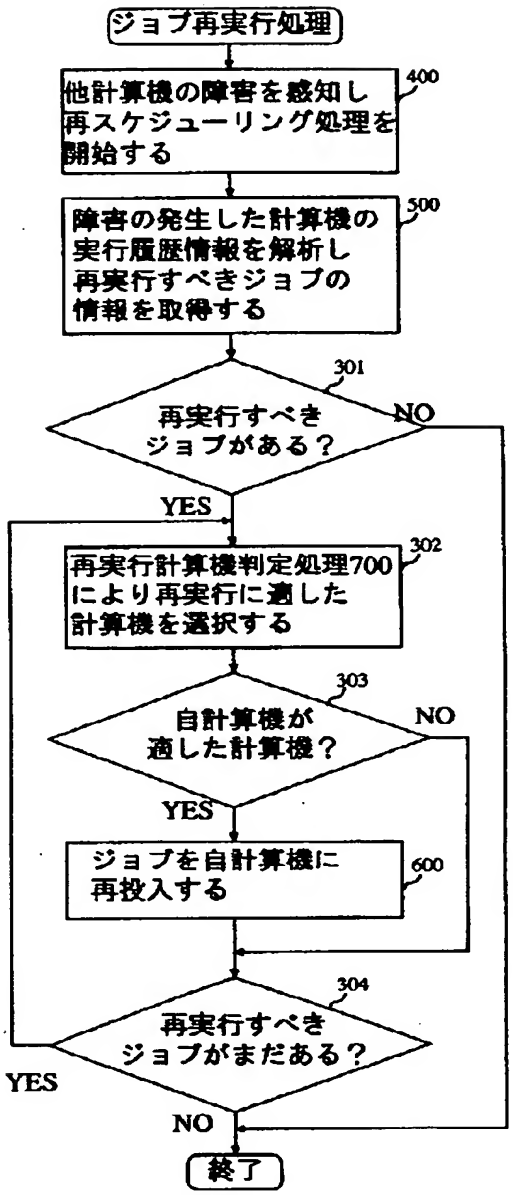
【図１】

図１


高速結合装置
C P U
102
103
101
メモリ
システム稼働情報
負荷情報
104
105
106-n
計算機
106-1
107-n
108-n
計算機
C P U
107-1
108-1
メモリ
300-1
ジョブ再実行部
400-1
障害認識部
500-1
ジョブ再実行情報取得部
600-1
ジョブ再投入部
再実行情報判定部
200-1
負荷情報制御部
801-1
負荷情報通知部
802-1
負荷情報更新部
800-1
ジョブ起動プログラム
109-1
ジョブ実行プログラム
110-1
600-n
ジョブ再投入部
700-n
ジョブ起動プログラム
ジョブ実行プログラム
ジョブ実行履歴
200
ジョブ登録簿
111
ジョブ制御文
112

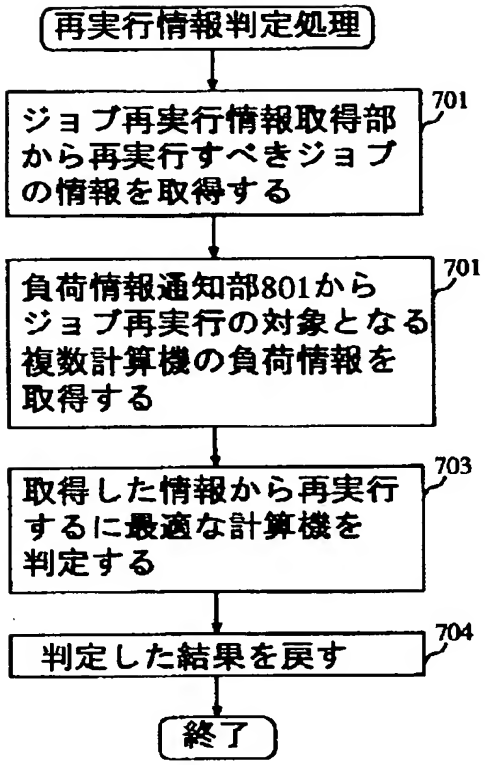
【図３】

図３


ジョブ再実行処理
他計算機の障害を感知し再スケジューリング処理を開始する
400
障害の発生した計算機の実行履歴情報を解析し再実行すべきジョブの情報を取得する
500
301
再実行すべきジョブがある？
NO
YES
再実行計算機判定処理700により再実行に適した計算機を選択する
302
303
自計算機が適した計算機？
NO
YES
ジョブを自計算機に再投入する
600
304
再実行すべきジョブがまだある？
YES
NO
終了

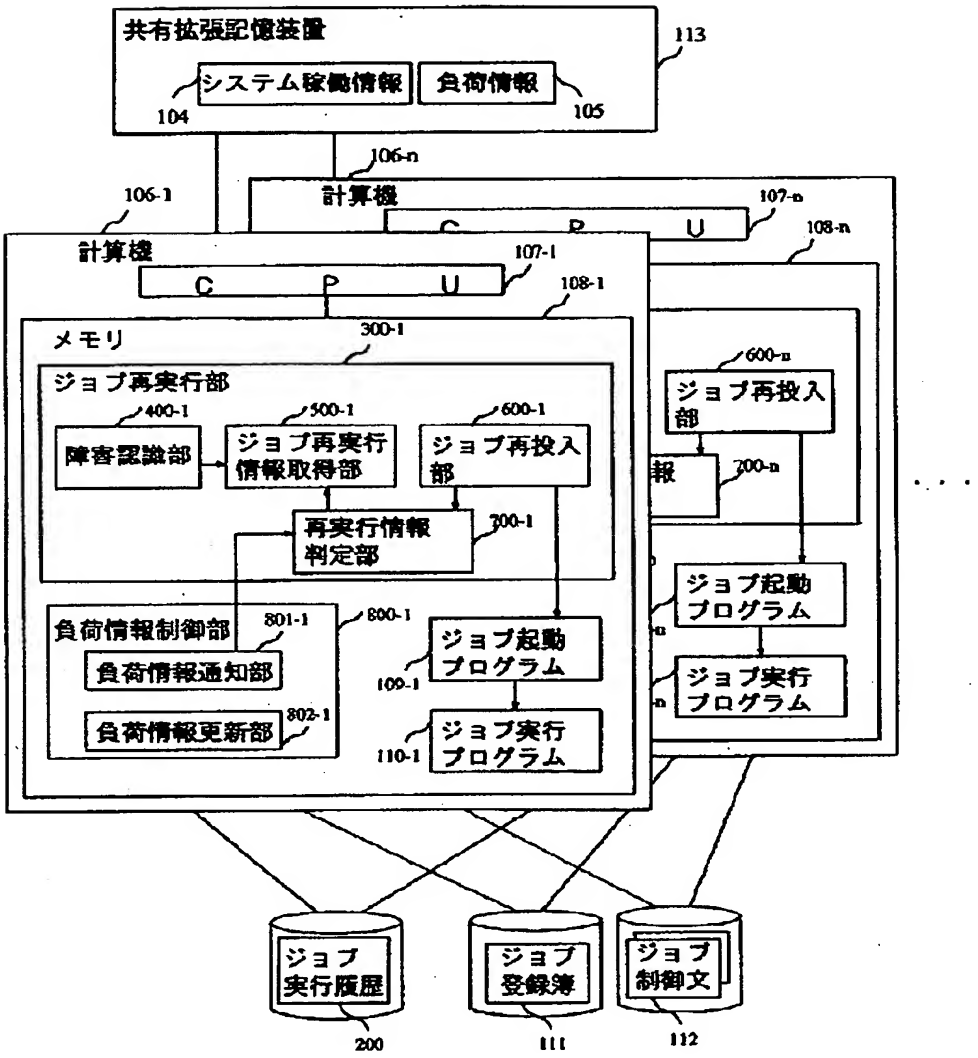
【図４】

図４


再実行情報判定処理
ジョブ再実行情報取得部から再実行すべきジョブの情報を取得する
701
負荷情報通知部801からジョブ再実行の対象となる複数計算機の負荷情報を取得する
701
取得した情報から再実行するに最適な計算機を判定する
703
判定した結果を戻す
704
終了

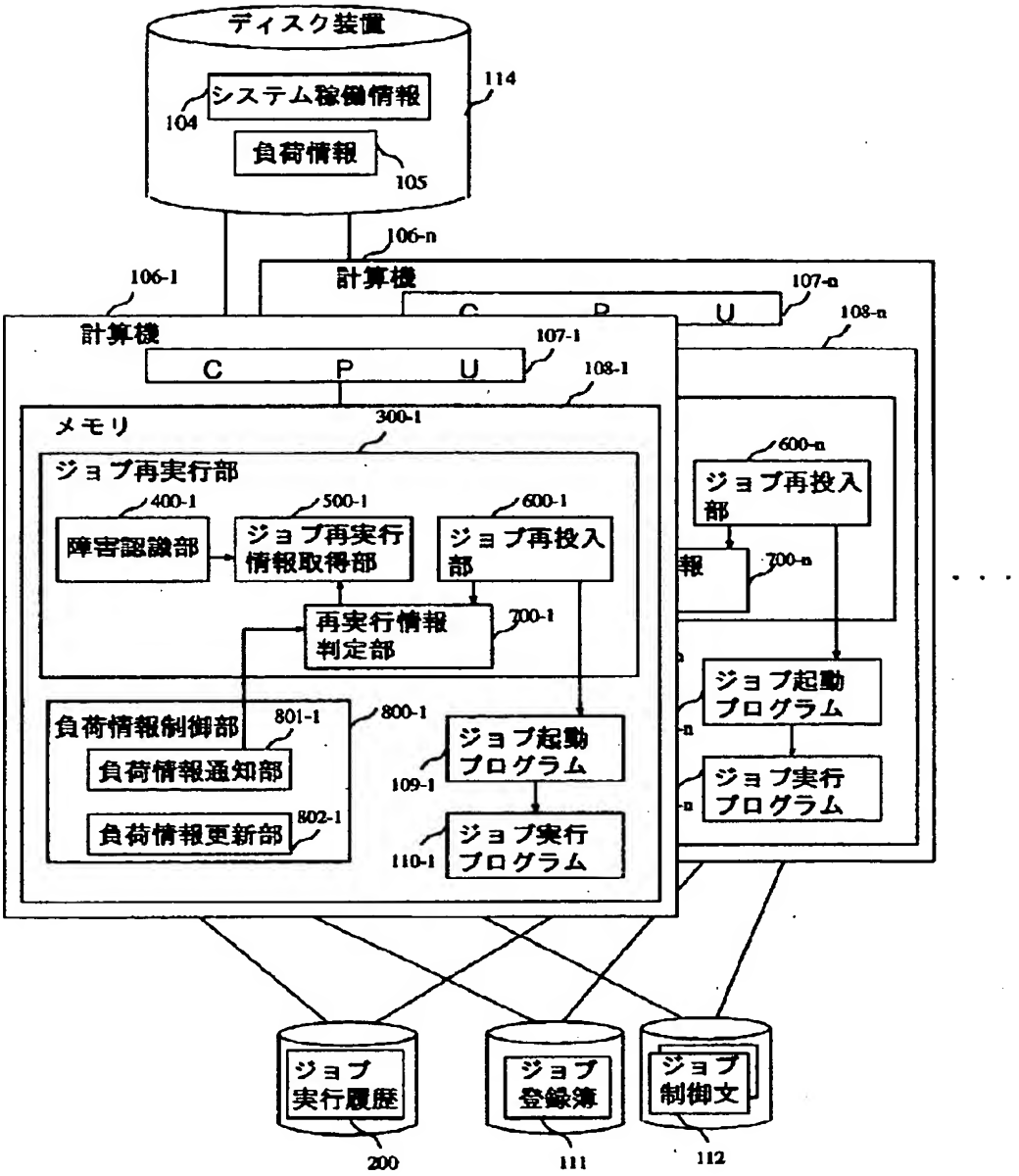
【図５】

図５


共有拡張記憶装置
113
システム稼働情報
104
負荷情報
105
106-n
計算機
107-n
C P U
108-n
106-1
計算機
107-1
C P U
108-1
メモリ
300-1
ジョブ再実行部
400-1
障害認識部
500-1
ジョブ再実行情報取得部
600-1
ジョブ再投入部
再実行情報判定部
700-1
負荷情報制御部
801-1
負荷情報通知部
802-1
負荷情報更新部
800-1
ジョブ起動プログラム
109-1
ジョブ実行プログラム
110-1
600-n
ジョブ再投入部
200-n
ジョブ起動プログラム
ジョブ実行プログラム
ジョブ実行履歴
200
ジョブ登録簿
111
ジョブ制御文
112

【図６】

図６



フロントページの続き

(72)発明者　岩倉　義之
神奈川県横浜市戸塚区戸塚町5030番地　株式会社日立製作所ソフトウェア開発本部内